

WHAT IS CLAIMED IS:

1. A method for reporting latency information as perceived by a client in a client server system, the method comprising:

 dispatching a first request from a client to a server;
 receiving a first response from the server, wherein the first response corresponds to the first request;

 measuring a latency from the client's dispatch of the first request to the client's receipt of the first response from the server;

 appending the latency information to a second request;
 dispatching the second request with the latency information from the client to the server.

2. The method of claim 1, wherein the client is a messaging client, and wherein the server is a messaging server.

3. The method of claim 1, wherein the first and second requests are remote procedure calls (RPCs).

4. The method of claim 1, further comprising accumulating latency information from the client at the server.

5. The method of claim 1, further comprising accumulating latency information for a plurality of clients at the server.

6. The method of claim 1, further comprising accumulating latency information for a plurality of clients and a plurality of servers at the server.

7. The method of claim 6, wherein each of the plurality of servers is a different type of messaging server.

8. The method of claim 7, wherein the plurality of servers include mail servers, public folder servers, and calendar/scheduling servers, and subcombinations thereof.

9. The method of claim 1, wherein the client is an e-mail client, wherein the server is an e-mail server.

10. A software program embodied on a computer readable medium, wherein the software program is executable to perform the method of claim 1.

11. A method for determining remote procedure call (RPC) performance in a client server system, the method comprising:

monitoring on a client a status of each of a plurality of RPCs sent to a server;

appending information regarding the status of the RPC to at least one subsequent RPC that is sent to the server;

generating aggregate statistics on the server regarding RPC performance of the server as perceived by the client based upon the status; and

generating an alert if the aggregate statistics exceed a particular threshold indicative of problematic RPC performance.

12. A computer-implemented method, comprising:
sending a first request from a client to a server;
recording, at the client, a request initiation time for the first request;
receiving, at the client, a first response from the server corresponding to the first request;
recording, at the client, a response received time for the first response;
calculating a round trip latency for the first request/response pair comprising a difference between the response received time for the first response and the request initiation time for the first request; and
sending a second request from the client to the server, the second request comprising performance data, and the performance data comprising the round trip latency for the first request/response pair.

13. The method according to claim 12, further comprising:
receiving the first request at the server;
recording, at the server, a request received time for the first request;
recording, at the server, a response initiation time for the first response;
calculating a server processing time for the first request/response pair comprising a difference between the

response initiation time for the first response and the request received time for the first request;
 sending the first response to the client;
 sending the server processing time for the first request/response pair to the client; and
 recording, at the client, the server processing time for the first request/response pair.

14. The method according to claim 13, wherein the first response comprises the server processing time for the first request/response pair.

15. The method according to claim 14, wherein the performance data further comprises the server processing time for the first request/response pair.

16. The method according to claim 15, further comprising:
 receiving the second request at the server;
 parsing the performance data from the second request;
and
 updating at least one computer system memory resident performance data accumulator with the performance data.

17. The method according to claim 16, wherein the at least one computer system memory resident performance data accumulator comprises:
 a request count accumulator;
 a request class count accumulator;
 a request rate accumulator;

a request class rate accumulator; and
a count of a performance data parameter value exceeding
a threshold accumulator.

18. The method according to claim 17, further
comprising generating an event log entry if the performance
data triggers a performance data event and the performance
data event would not contribute to a performance data event
storm.

19. The method according to claim 18, wherein
triggering a performance data event comprises:
a ratio of successful requests to total requests
dropping below a minimum; and
an average request latency rising above a maximum.

20. The method according to claim 18, wherein
contributing to a performance data event storm comprises
generating more than a maximum number of performance data
events in a particular performance data event class during a
period of time.

21. The method according to claim 12, wherein the
performance data further comprises client server
communications session invariant performance data context,
and the performance data context comprising at least one
performance data context identifier.

22. The method according to claim 21, wherein the
performance data context further comprises:

- a client computer system host name;
- a client user name;
- a client network adaptor name;
- a client network adaptor speed; and
- a client network protocol address; and
- subcombinations thereof.

23. The method according to claim 22, wherein the performance data context further comprises:

- a server computer system host name;
- a server network domain name; and
- a server type; and
- subcombinations thereof.

24. The method according to claim 23, wherein the performance data context further comprises:

- a globally unique identifier (GUID) associated with a client computer operating system process; and
- information regarding the client computer operating system process; and
- subcombinations thereof.

25. The method according to claim 24, wherein the performance data context further comprises a globally unique identifier (GUID) associated with a client server communications session.

26. The method according to claim 12, wherein the second request further comprises an indication of at least one service desired of the server by the client.

27. A computer-readable medium having thereon computer executable instructions for performing the method according to claim 12.

28. A computer-implemented method, comprising:
sending a first request from a client to a server;
recording, at the client, an error condition
corresponding to the first request; and
sending a second request from the client to the server,
and the second request comprising:
an indication of at least one service desired of
the server by the client; and
performance data, and the performance data
comprising the error condition corresponding to the first
request.

29. The method according to claim 28, further
comprising:
receiving the second request at the server;
parsing the performance data from the second request;
classifying the first request based on the error
condition corresponding to the first request; and
updating at least one computer system memory resident
performance data accumulator associated with the request
class.

30. A computer-implemented method, comprising:
sending a first request from a client to a first server;

recording, at the client, a request initiation time for the first request;

receiving, at the client, a first response from the first server corresponding to the first request;

recording, at the client, a response received time for the first response;

calculating a round trip latency for the first request/response pair comprising a difference between the response received time for the first response and the request initiation time for the first request; and

sending a second request from the client to a second server, the second request comprising performance data, and the performance data comprising the round trip latency for the first request/response pair.

31. The method according to claim 30, further comprising:

receiving the second request at the second server;

parsing the performance data from the second request;

and

updating, with the performance data, at least one computer system memory resident performance data accumulator associated with the first server.

32. A computer-implemented method, comprising:

sending a first request from a client to a server;

recording, at the client, a request initiation time for the first request;

receiving, at the client, a first response from the server corresponding to the first request;

recording, at the client, a response received time for the first response;

calculating a round trip latency for the first request/response pair comprising a difference between the response received time for the first response and the request initiation time for the first request;

storing, at the client, performance data associated with the first request/response pair and a performance data storage time, the performance data comprising the round trip latency for the first request/response pair;

sending a second request from the client to the server;
and

if a difference between a request initiation time for the second request and the storage time for the performance data associated with the first request/response pair is less than a maximum performance data age threshold then incorporating the performance data associated with the first request/response pair into the second request.

33. The method according to claim 32, further comprising receiving the maximum performance data age threshold from the server.

34. A computerized client system, comprising:

a performance data store;

a performance data measurement module configured to, at least:

generate performance data concerning requests from the client system and corresponding responses to the client system; and

store the generated performance data in the performance data store;

a server performance data preference store;

a performance data stream parse module configured to, at least, parse an incoming data stream for server performance data preferences and store them in the server performance data preference store;

a performance data stream format module configured to, at least, format and insert performance data from the performance data store into an outgoing data stream in accordance with server performance data preferences.

35. The system of claim 34, further comprising:

a performance data context map capable of maintaining, for each performance data context, a performance data context identifier to performance data context association;

a performance data contextualize module configured to, at least:

 instantiate at least one performance data context;

and

 for each performance data context, make a performance data context identifier to performance data context association entry in the performance data context map.

36. The system of claim 35, wherein the performance data stream format module is further configured to, at least:

 format and insert performance data context independently from other performance data into an outgoing data stream; and

replace performance data context with its associated performance data context identifier, as specified in the performance data context map.

37. The system of claim 34, wherein the incoming data stream is buffered by the client system in a buffer with a fixed size, and wherein the outgoing data stream is buffered by the client system in a buffer with a fixed size.

38. A computerized server system, comprising:

- a performance data stream parse module configured to, at least, parse client-generated performance data from an incoming data stream;

- at least one server system memory resident performance data accumulator; and

- a performance data report module configured to, at least, update the at least one server system memory resident performance data accumulator from performance data corresponding to the parsed client-generated performance data.

39. The system of claim 38, wherein the client-generated performance data comprises performance data context, the performance data stream parse module is further configured to, at least, parse performance data context from the incoming data stream, and the system further comprises:

- a performance data context map capable of maintaining, for each parsed performance data context, a performance data context identifier to performance data context association; and

a performance data contextualize module configured to, at least, for each parsed performance data context, make a performance data context identifier to performance data context association entry in the performance data context map.

40. The system of claim 38, further comprising:
a server performance data preference store; and
a performance data stream format module configured to, at least, format and insert server performance data preferences from the server performance data preference store into an outgoing data stream.

41. The system of claim 38, further comprising:
a performance data event log;
a performance data event trigger database comprising at least one criteria set corresponding to criteria for triggering a performance data event; and
wherein the performance data report module is further configured to, at least:
for each criteria set in the performance data event trigger database, determine if the criteria set is met as a consequence of the parsed client-generated performance data; and
for each criteria set that is met:
generate a performance data event; and
enter the performance data event in the performance data event log.

42. A computer-readable medium having stored thereon a data structure, comprising:

a performance data remote procedure call (RPC) extension tag, comprising:

a performance data format version field;

at least one performance data format flag;

a performance data size field; and

an uncompressed performance data size field; and

at least one performance data block, comprising:

a performance data block header, comprising:

a performance data block size field;

a performance data block format version field;

and

a performance data block type field; and

a performance data block body.

43. The computer-readable medium of claim 42, wherein the at least one performance data format flag is selected from a group consisting of compress performance data and obfuscate performance data.

44. The computer-readable medium of claim 42, wherein the performance data block body comprises:

a variable byte array storage area; and

at least one string type field comprising an offset to a location in the variable byte array storage area where the value of the string type field is stored.

45. The computer-readable medium of claim 42, wherein the performance data block type field indicates that the

performance data block body contains client information performance data context pertaining to a client in a client server network, and wherein the performance data block body comprises:

- a client-generated client information performance data context identifier;
- a name of a computer system hosting the client;
- a name of a user utilizing the client;
- a network protocol address associated with the client;
- a name of a network adapter associated with the client;

and

- a speed of the network adapter associated with the client.

46. The computer-readable medium of claim 42, wherein the performance data block type field indicates that the performance data block body contains server information performance data context pertaining to a server in a client server network, and wherein the performance data block body comprises:

- a client-generated server information performance data context identifier;
- a name of the server;
- a network domain name of the server; and
- a server type associated with the server.

47. The computer-readable medium of claim 42, wherein the performance data block type field indicates that the performance data block body contains client-generated performance data pertaining to a successful request/response

pair between a client and a server in a client server network, and wherein the performance data block body comprises:

- a client-generated request identifier associated with the successful request/response pair;

- a round trip latency corresponding to a difference between a time at which the client initiated the successful request/response pair and a time at which the client received the response; and

- a server processing time corresponding to a difference between a time at which the server received the request of the successful request/response pair and a time at which the server initiated the response.

48. The computer-readable medium of claim 47, wherein the performance data block size is 14 bytes.

49. The computer-readable medium of claim 42, wherein the performance data block type field indicates that the performance data block body contains client-generated performance data pertaining to a failed request/response pair between a client and a server in a client server network, and wherein the performance data block body comprises:

- a client-generated request identifier associated with the failed request/response pair;

- a time to fail corresponding to a difference between a time at which the client initiated the failed request/response pair and a time at which the client determined that the request had failed; and

a failure code corresponding to a reason for the failure of the request.

50. The computer-readable medium of claim 42, wherein the performance data block type field indicates that the performance data block body contains server performance data preference information for a server in a client server network, and wherein the performance data block body comprises:

- an indication of whether to send client-generated performance data to the server pertaining to client communications with the server;

- an indication of whether to send client-generated performance data to the server pertaining to client communications with other servers in the client server network; and

- a performance data age threshold beyond which client-stored performance data should not be sent to the server.